

**AMENDMENTS TO THE CLAIMS:**

The following listing of claims supersedes all prior versions and listings of claims in this application:

1. (Currently Amended) A computer system comprising: having  
a plurality of components that can be initialized, each component being operable  
to perform at least one task when not being initialized.

wherein each component is configured to produce status data from which the level of need for that component to be initialized, can be inferred, the status data representing having a predetermined level of need for that component to be initialized  
~~associable therewith~~, and

wherein at least one component is configured to:

receive status data from other components;

make a comparison using the status data received from respective  
components;

in dependence on the comparison, select one or more components for  
initialization; and[[,]]

issue initialization instructions to the selected component(s).

2. (Currently Amended) A computer system as claimed in claim 1, wherein the ~~of~~ each component configured to make a comparison using status data is configured to use its own status data in addition to ~~the~~ received status data when making the comparison.

3. (Previously Presented) A computer system as claimed in claim 1, wherein the components are software components, and wherein the system includes at least one computer device on which, in use, the software components are run.

4. (Previously Presented) A computer system as claimed in claim 1, wherein the status data is in the form of an initialization parameter.

5. (Currently Amended) A computer system as claimed in claim 4, wherein:  
each component is configured to execute an initialization routine when the initialization parameter for that component reaches a respective threshold value,  
the initialization routine including the step of transmitting a request for an initialization parameter to other components.

6. (Currently Amended) A computer system as claimed in claim 5, wherein the initialization routine includes the further steps of:

receiving initialization parameters from at least some of those other components;  
comparing the received initialization parameters with the initialization parameter  
for that component; and[[,]]  
in dependence on the comparison, making a self- initialization decision.

7. (Currently Amended) A computer system as claimed in claim 4, wherein;  
each component includes a timer module for registering the elapsed time since  
the previous initialization of that component, and ~~wherein~~  
for each component, the initialization parameter is determined at least in part in  
dependence on the elapsed time registered by the timer module.

8. (Previously Presented) A computer system as claimed in claim 4, wherein  
each component is configured to produce an initialization parameter that is at least in  
part dependent on whether the component is performing one of a number of  
predetermined tasks.

9. (Previously Presented) A computer system as claimed in claim 3, wherein the  
computer system includes a plurality of interconnected computer devices, each of which  
is housed in a respective housing, and wherein each device has, in use, a respective  
software component running thereon.

10. (Original) A computer system as claimed in claim 9, wherein the software components each include a respective operating system module for operating the computer device on which the respective software component is running.

11. (Previously Presented) A computer system as claimed in claim 9, wherein each component is configured to initiate a re-boot routine upon receipt of an initialization instruction, the re-boot routine being configured to re-boot the computer device on which the software component is running.

12. (Original) A computer system as claimed in claim 11, wherein the re-boot routine includes the step of determining if the computer device is performing a predetermined task or one of a number of predetermined tasks, and only to permit the re-booting of the computer device if the computer device is not performing such a task.

13. (Previously Presented) A computer system as claimed in claim 3, wherein the components in use run on a common computer device, under the control of a common operating system.

14. (Previously Presented) A computer system as claimed in claim 3, wherein each component, upon receipt of an initialization instruction, is configured such that the component is killed and subsequently restarted.

15. (Previously Presented) A computer system as claimed in claim 1, including a computer device configured to allocate tasks to the components, such that a task allocated to one component is dependent on the task or tasks being performed by at least some of the other components.

16. (Currently Amended) A method of initializing the components of a computer system, the method comprising ~~including the steps of:~~

receiving status data from a plurality of components, each component being operable to perform at least one task when not being initialized, the received status data having associable therewith a predetermined level of need for a component to be initialized;

determining for each component, the need for that component to be initialized relative to the need for at least one other component to be initialized; and[[,]]

initializing at least some of the components in dependence on their so determined relative need.

17. (Currently Amended) A computer device having, in use, a software component running thereon, the software component being configured to:

receive status data from a plurality of other components, each component being operable to perform at least one task when not being initialized, the received status data having associable therewith a predetermined level of need for a component to be initialized;

make a comparison using the status data received from the respective components;

in dependence on the comparison, select one or more components for initialization; and[[,]]

issue initialization instructions to the selected component(s).

18. (Previously Presented) A computer device as claimed in claim 17, wherein the software component running thereon is configured to produce status data from which the need for that component to be initialized can be inferred.

19. (Previously Presented) A computer device as claimed in claim 18, wherein the software component is configured to compare its self-produced status data with status data received from at least one other component when selecting one or more components for initialization.

20. (Currently Amended) A computer device ~~readable storage medium~~ containing a computer program, the computer program being configured, when executed by loaded on a computer device processor, to:

receive status data from a plurality of software components, the received status data having associable therewith a predetermined level of need for a component to be initialized;

make a comparison using the received status data from the respective components;

in dependence on the comparison, select one or more of the software components for initialization; and[[,]]

issue initialization instructions to the selected software component(s).

21. (Currently Amended) A computer device ~~readable storage medium~~ as claimed in claim 20, wherein the computer program is configured to issue initialization instructions for re-booting the computer device or another computer device.

22. (Currently Amended) A computer product containing a computer program ~~product~~ stored on a computer-readable medium, the computer program product having a computer program stored thereon as claimed in claim 20.

23. (Currently Amended) A computer system comprising: having  
a plurality of components that can be initialized, each component being operable  
to perform at least one task when not being initialized.

wherein each component is configured to produce status data from which the  
level of need for that component to be initialized can be inferred, and

wherein at least one component is configured to:

receive status data from other components;

make a comparison using the status data received from respective  
components;

in dependence on the comparison, select one or more components for  
initialization; and[[,]]

issue initialization instructions to the selected component(s).

24. (Currently Amended) A method of initializing the components of a computer  
system, each component being operable to perform at least one task when not being  
initialized. the method comprising ~~including the steps of~~:

using status data from a plurality of components to determine for each  
component, the need for that component to be initialized relative to the need for at least  
one other component to be initialized; and[[,]]



initializing at least some of the components in dependence on their so  
determined relative need.